

Programming project- Table tennis management system

Student 1: Kis Ariana

Student 2: Iftinca Lucas

1. Task description

Student 1 is tasked with managing matches:

- add new matches
- save them

Student 2 is tasked with viewing data:

- player stats
- leaderboards
- matches

2. Data Structures Used by the Team

The following classes will be used:

- **Player:**
 - string name — The player's name (unique identifier).
 - int wins — Number of matches won.
 - int matches — Total matches played.

- **Match:**
 - string pl1_name — Player 1's name.
 - int pl1_score — Player 1's score.
 - string pl2_name — Player 2's name.
 - int pl2_score — Player 2's score.
 - *Note:* Each match records the competing players and their respective scores.
- **Date** (optional, if you want to track when matches happen):
 - int day
 - int month
 - int year
- **Leaderboard** (conceptual):
 - vector<Player> players — A collection of players sorted by wins or matches.

3. File structure

The project will consist of the following files:

Data Files

- **match_info.txt**
Stores match records in the format:

<player1_name>,<player1_score>,<player2_name>,<player2_score>

Each line represents one match between two players.

Source Files

- **match.hpp**
Header file declaring the Match class and its members (e.g., constructors, isValid(), writeToFile(), readAll()).
- **match.cpp**
Implementation of the Match class functions declared in match.hpp.
- **player.hpp**
Header file declaring the Player class and its members (e.g., name, wins, matches, static functions for sorting and generating players from matches).

- **player.cpp**
Implementation of the Player class functions declared in player.hpp.
- **utils.hpp**
Header file declaring utility functions like valid_name(), valid_score(), and any helper functions.
- **utils.cpp**
Implementation of the utility functions declared in utils.hpp.
- **ui.hpp**
Header file declaring UI-related functions like main_menu(), choice(), and others related to input/output display.
- **ui.cpp**
Implementation of the UI functions declared in ui.hpp.
- **main1.cpp**

The first executable that contains the main entry point dedicated to adding a new match. It handles user input, validates names and scores, ensures match rules are respected, and saves the valid match to the file system.**main2.cpp**
Contains the main entry point for the program, handles the program loop, calls UI functions, and coordinates user interaction.
- **main2.cpp**

The second executable that includes the main entry point for all other program features, such as viewing match history, checking individual player statistics, and displaying leaderboards sorted by wins or matches played. It provides a user-friendly interface for navigating these options.

4. Interacting with executables

Application 1 – Match Management

Executable name: app_1.exe

Purpose: This application allows users to manage match records. The primary feature is adding new matches to the system.

Available Commands:

- **app_1.exe new_match**

Opens an interactive prompt to input details for a new match:

- Player 1 name
- Player 1 score
- Player 2 name
- Player 2 score

The input is validated, and if valid, the match is saved in match_info.txt.

Application 2 – Leaderboard and Statistics

Executable name: app_2.exe

Purpose: This application is used to view the current state of matches, player statistics, and leaderboards.

Available Commands:

- **app_2.exe view_matches**

Displays all saved matches, highlighting the winner using colors.

- **app_2.exe player_stats <player_name>**

Shows the number of wins, losses, and total matches for a specific player.

- **app_2.exe leaderboard_wins**

Displays a leaderboard sorted by the most number of wins.

- **app_2.exe leaderboard_matches**

Displays a leaderboard sorted by the most matches played.

